# Which metrics to use for RF indirect test strategy?

Hassan El Badawi, Mariane Comte, Florence Azaïs, Vincent Kerzérho, Serge Bernard, François Lefevre

# Which metrics to use for RF indirect test strategy?

Hassan El Badawi
*LIRMM,*
*University of Montpellier, CNRS*
Montpellier, France
elbadawi@lirmm.fr

Florence Azais
*LIRMM,*
*University of Montpellier, CNRS*
Montpellier, France
azais@lirmm.fr

Serge Bernard
*LIRMM,*
*University of Montpellier, CNRS*
Montpellier, France
bernard@lirmm.fr

Mariane Comte
*LIRMM,*
*University of Montpellier, CNRS*
Montpellier, France
comte@lirmm.fr

Vincent Kerzérho
*LIRMM,*
*University of Montpellier, CNRS*
Montpellier, France
kerzerho@lirmm.fr

François Lefevre
*NXP Semiconductors*
Caen, France
francois.lefevre@nxp.com

***Abstract*—This paper aims at opening a discussion on the quality assessment of indirect test strategies in the context of analog and RF integrated circuit testing. Many parameters may influence the prediction efficiency of the indirect test model (choice and number of indirect parameters taken into account, learning algorithm used to build the model…). In order to evaluate the quality of a given model, several metrics can be evaluated, that reflect either the average prediction error, a global reliability or a misclassification rate. But what are the most pertinent metrics to reflect the level of confidence that can be expected from the indirect test to efficiently replace a traditional test based on RF measurements? Which metrics can lead to an informed choice of an indirect test strategy for its stability and predictive power? These considerations are investigated in this paper and illustrated in a practical case study.***

***Keywords*— RF integrated circuits, indirect testing, machine-learning, metrics, test efficiency**

## I. Introduction

Several manufactured Integrated Circuits (IC) do not meet the targeted product specifications. Indeed, process variations and/or physical defects can degrade the performance of a circuit, or even drastically affect its operation. It is therefore essential to test the performance of each circuit produced before providing it or integrating it into a more complex system. The testing process represents a significant part of the total cost of an IC, especially for analog and RF circuits, whose performance must be measured with sophisticated and expensive test equipment. In order to reduce testing costs, one possible strategy is to adopt indirect testing, which consists in measuring parameters that require only low-cost test resources and correlating these measurements, called Indirect Measurements (*IMs*), with the device specifications. This correlation is generally established using machine-learning algorithms during an initial training phase. This approach has been introduced first for analog circuits [1], and then extended to RF circuits [2]. Several aspects have been researched, such as the influence of the training set [3], the use of embedded sensors to gather pertinent information [4], the exploitation of multi-Vdd test conditions [5], or the selection of appropriate indirect measurements [6-9]. A comprehensive review of works related to indirect testing can be found in [10].

While the indirect test strategy seems attractive, its deployment in an industrial context is viable only if sufficient test quality can be achieved. However, it’s extremely difficult to assess at the learning phase what the test coverage will achieve during the industrial production test. Moreover, there is no general consensus on what is a pertinent and objective metric that allows the comparison of various model constructions in terms of indirect test efficiency. It is the objective of this paper to analyze and discuss these aspects, based on a practical case study.

This paper is organized as follows. Section II summarizes the basics of the indirect test approach and the experimental protocol used for model construction and evaluation. Section III is devoted to the definition of the metrics commonly used in the context of indirect testing. Finally, results obtained on a practical case study are presented in Section IV.

## II. Indirect Test Strategy

### A. Indirect Test Principle

The underlying idea of indirect testing is that process variations that affect the device performance also affect indirect parameters. If the correlation between the indirect parameter space and the specification space can be established, then specifications may be verified using only the low-cost indirect signatures. Unfortunately, the relation between these two sets of parameters is complex and cannot be simply identified with an analytic function. The solution commonly implemented is based on the use of machine-learning algorithms. The indirect test synopsis is actually split into two distinct phases, i.e. test preparation and production test, as illustrated in Figure 1.

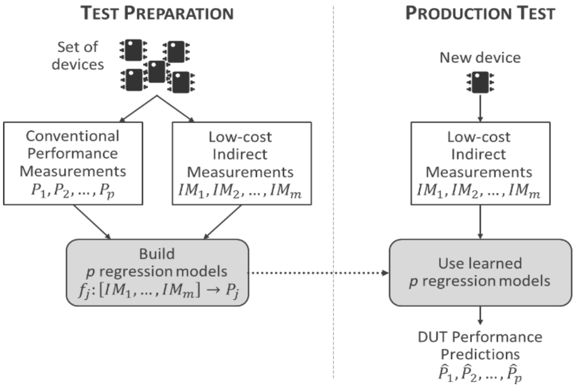
TEST PREPARATION
PRODUCTION TEST
Set of devices
New device
Conventional Performance Measurements $P_1, P_2, \ldots, P_p$
Low-cost Indirect Measurements $IM_1, IM_2, \ldots, IM_m$
Low-cost Indirect Measurements $IM_1, IM_2, \ldots, IM_m$
Build $p$ regression models $f_j: [IM_1, \ldots, IM_m] \to P_j$
Use learned $p$ regression models
DUT Performance Predictions $\hat{P}_1, \hat{P}_2, \ldots, \hat{P}_p$

Fig.1. Indirect test synopsis.

The objective of the initial test preparation phase is to build regression models that map the indirect parameters space to the performance parameters space. In this phase, both the specification tests and the low-cost measurements are performed on a set of devices. These data are then fed to a machine-learning algorithm, which is trained to learn the dependency between the indirect measurements and the conventional ones. Once the training is completed, the mass production testing phase can start. In this phase, only the indirect measurements are performed, and the specifications of every new device are predicted using the mapping learned

### B. Experimental Protocol

A key element for the success of an indirect test is the construction of efficient regression models during the initial test preparation phase. This is not an easy task as many different solutions are possible, in particular regarding the choice of appropriate indirect measurements (*IMs*) and the choice of the learning algorithm. A common practice is therefore to explore different options during the test preparation phase and to retain the most performing one. Figure 2 gives a synthetic view of the experimental protocol generally employed.

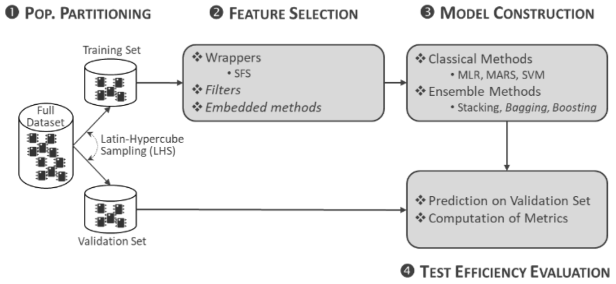

Fig.2. General overview of the experimental protocol.

It consists of 4 main phases. The first phase involves the partitioning of the population into two different sets. The first one will be used to train the prediction model and the second one to evaluate the constructed model. Note that it is important to evaluate the performance of the model on different instances than the ones used for training, to verify the generalization ability of the model and avoid issues related to overfitting. A simple way to partition data is to use random sampling. However, such sampling method does not guarantee that training and validation sets have similar statistical characteristics which can reduce the model's estimation accuracy. A more refined technique is to use Latin Hypercube Sampling (*LHS*), which ensures that the generated sets are representative of the real variability. This is the solution we have implemented in this work.

The second phase consists in selecting pertinent *IMs* among the set of available measurements. Indeed, the construction of a model that uses all available *IMs* will inevitably suffer from overfitting. Moreover, the use of a limited number of *IMs* contributes to the reduction of testing costs. This problem of selecting a subset of features among a larger set is a recurrent problem in the field of machine-learning, known as feature selection. Various algorithms have been proposed, which can be divided into three categories, namely filters, wrappers and embedded methods [11]. In the context of indirect testing, the solution commonly employed is a wrapper method based on Sequential Forward Selection (SFS). This procedure starts with an empty set and sequentially construct models by adding the feature that minimizes the prediction error when combined with the features that have already been selected. In this work, we have implemented such a procedure with a maximum number of 15 features.

The third phase consists in building a regression model using the features selected in the previous phase. The classical approach is to build a single regression model for each performance to be predicted. Many different algorithms exist to perform this task; the most popular algorithms used in the context of indirect test are Multiple Linear Regression (*MLR*), Multi-Adaptive Regression Splines (*MARS*), and Support Vector Machine (*SVM*). These three algorithms have been implemented in this work. An alternative approach is to build multiple regression models for each performance to be predicted and aggregate their outcomes to get the final prediction results. The idea is that with an appropriate combination of diverse individual models, it should be possible to exploit the strengths and overcome the weaknesses of the individual models and obtain better stability and predictive power. This approach is called ensemble learning and numerous methods for constructing ensemble models have been proposed in the literature [12]; the most popular methods are bagging, boosting and stacking. Basically, bagging and boosting methods rely on a manipulation of the training data in order to build multiple base learners using a single model type. In contrast, stacking relies on the use of different model types to build multiple base learners; the outputs of these base learners are then used to train a higher-level learner, called meta-learner. The use of ensemble method for test application, has been firstly introduced in [5]. A recent work has shown the superiority of such models compared with bagging or boosting in the context of indirect testing [13]. Hence, we have implemented in this work the construction of an ensemble model based on stacking. More precisely, 3 different base learners have been trained using *MLR*, *MARS* and *SVM*; their outputs have been then used to train the meta-learner using *MARS*.

Finally, the last phase concerns the evaluation of the test efficiency in order to retain the most efficient solution. In this phase, all the models built in the previous phase are used to perform prediction of devices of the validation set. Several metrics are then computed to evaluate the performance of the different models. But what are the most pertinent metrics to perform this evaluation? This is the key question that we target in this paper. The following section is therefore devoted to the description of the main metrics used in the context of indirect testing.

## III. Metrics for Indirect Test Efficiency Evaluation

The most commonly-used metric to evaluate the accuracy of a model is the Root Mean Square Error (*RMSE*). It corresponds to a measure of the differences between actual and predicted values. Basically, the *RMSE* corresponds to a measure of the average prediction error and is expressed in units of the variable of interest; its value is therefore dependent of the variable scale.

To facilitate the comparison between datasets or models with different scales, the Normalized Root Mean Square Error (*NRMSE*) is also commonly used:

$$NRMSE = \frac{\sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i-\hat{y}_i)^2}}{\bar{y}} \quad (1)$$

where $y_i$ is the actual performance value of the i^th instance, $\hat{y}_i$ is the predicted performance value of the i^th instance, $n$ is the number of instances in the validation set and $\bar{y}$ is the mean of the observed data.

The prediction error is in this case expressed in percentage. Globally, the lower the *NRMSE* (or the *RMSE*), the better the accuracy of the model.

The problem with this metric is that it gives an image of the overall quality of a model, but it doesn't give any information on the how the prediction errors are distributed. In particular, it doesn't give any information on whether all circuits are predicted with a similar prediction error or whether some circuits are predicted with a low prediction error but some others with a large prediction error. Moreover, the prediction errors are computed assuming that the actual values are perfectly known, which is obviously an invalid assumption since the values determined with a conventional specification test are necessarily subjected to a measurement uncertainty.

In this context, another metric has been introduced in [14] which permits to evaluate the quality of a model in terms of its reliability with respect to the measurement uncertainty of the conventional specification test. This metric, called Failing Prediction Rate (*FPR*), expresses the percentage of circuits with a prediction error that exceeds the conventional measurement uncertainty $\varepsilon_{meas}$ and is computed with:

$$FPR = \frac{1}{n} \sum_{i=1}^{n} (|y_i - \hat{y}_i| > \varepsilon_{meas}) \qquad (2)$$

with $(|y_i - \hat{y}_i| > \varepsilon_{meas}) = 1$ if true
$(|y_i - \hat{y}_i| > \varepsilon_{meas}) = 0$ otherwise

Both these metrics are devoted to the evaluation of the quality of a regression model, but they cannot be directly related to the indirect test efficiency. The common metric used in the literature to quantify the indirect test efficiency is the Misclassification Rate (*MR*). To compute this metric, the test limits must be known. Based on these test limits, all instances of the validation set are classified as good and bad circuits using the actual performance values on one hand and using the predicted values on the other hand. Both classifications are then compared and the number of incorrect decisions when using the predicted values is recorded. The Misclassification Rate expresses the percentage of circuits that have incorrect decisions among the total number of evaluated circuits.

The main issue with this metric is that it does not take into account the measurement uncertainty that affects the conventional specification test. The original classification performed using measured performance values is therefore questionable. Indeed because of the measurement uncertainty, it exists a region around the test limit where it's not possible to fully guarantee the classification (cf. Fig.3). More precisely, all circuits that have a measured value within this region might be either good or bad circuits; only circuits that have a measured value outside this region can be trustfully defined as good or bad circuits.

In this context, we propose to compute another metric that might be more representative of the indirect test efficiency, called Trusted Misclassification Rate (*T-MR*). The idea is to evaluate a misclassification rate based only on trusted classifications, i.e. to compute the percentage of circuits that have an incorrect decision with the indirect prediction among the number of circuits that have a certain decision with the conventional measurement.

In this work, we have implemented the computation of all these metrics and results obtained on a practical case study are discussed in the following section.

## IV. Results

The test vehicle is a LNA for which we have production test data on 3,850 devices. Data include on the one hand the measurement of the third-order intercept point (*IP3*), and on the other hand 79 low-cost indirect measurements which correspond to DC voltages on internal nodes (the device is equipped with an internal DC bus) and DC signatures delivered by built-in process monitors. Figure 3 illustrates the distribution of the *IP3* performance on the population of available samples. It's a non-Gaussian distribution with an excursion between 32dBm and 36dBm. The test limit for this specification is set at 34dBm and the measurement uncertainty is 0.5dB. The objective is to develop an indirect test solution for the prediction of *IP3* performance and to evaluate the efficiency that can be achieved.

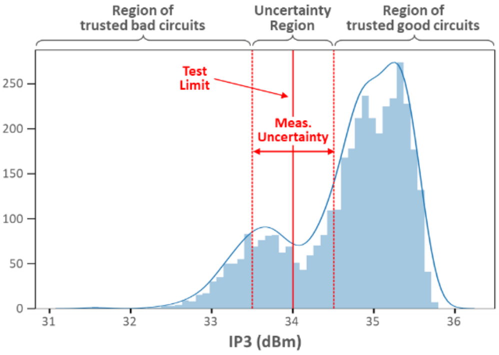

Fig.3. *IP3* distribution for the case study under investigation

The experimental protocol described in Section II has been applied to this case study. The data set has been partitioned into a training set composed of 2,000 devices and a validation set composed of 1,850 devices. The composition of the validation set is summarized in Table I.

TABLE I. Composition of the validation set

<table>
<tr><td colspan="6">Classification w.r.t. conventional IP3 specification test</td></tr>
<tr><td colspan="3">Good circuits</td><td colspan="3">Bad circuits</td></tr>
<tr><td colspan="3">1452</td><td colspan="3">398</td></tr>
<tr><td colspan="6">Classification w.r.t. conventional <i>IP3</i> specification test<br>taking into account measurement uncertainty</td></tr>
<tr><td colspan="2">Trusted good circuits</td><td colspan="2">Uncertain classifications</td><td colspan="2">Trusted bad circuits</td></tr>
<tr><td colspan="2">1270</td><td colspan="2">400</td><td colspan="2">180</td></tr>
</table>

The training set has been used to build four different types of regression models, i.e. three classical models (*MLR*, *MARS* and *SVM*) and one ensemble model based on stacking, varying the number of features from 1 up to 15. The validation set has then been used to compile the different metrics presented in Section III for all constructed models. Results are summarized in Figure 4, which reports the evolution of the metrics with respect to the number of features. Several comments arise from the analysis of these graphs.

First regarding the quality of the constructed models in terms of accuracy, it should be highlighted that it is possible to reach a very good accuracy for this case study. Indeed, for the four types of model, a low average prediction error is achieved, with a *NRMSE* below 1%. A slight advantage can be observed for *SVM* and *Stack* models, especially when only a limited number of features are used. The model with the least accuracy is as expected the simplest model, i.e. the *MLR* model. Globally according to this metric, there is no significant difference between the different types of model, with a *NRMSE* that ranges between 0.55% and 0.72% when a sufficient number of features are used.

Moreover, according to the FPR metric, which concerns the quality of the constructed models in terms of reliability, the same trends can be observed as in the case of model accuracy. Indeed, the most performing models are *SVM* and *Stack* models and the least performing ones are *MARS* and *MLR* models. However, the difference is this time more noticeable, with a best *FPR* around 0.6% and 0.7% for *SVM* and *Stack* models, and a best *FPR* around 1.2% and 1.5% for *MARS* and *MLR* models.

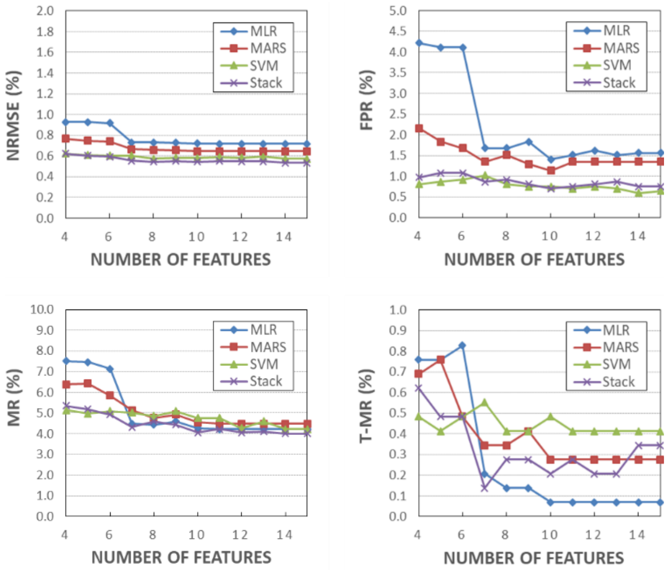

Fig.4. Evolution of metrics with respect to number of features, for different types of model

Regarding the indirect test efficiency in terms of misclassification rate, the situation is somehow different. Indeed, when a sufficient number of features is used, there is no clear evidence of the superiority of *SVM* and *Stack* models; the situation is even opposite with the *MLR* model that tends to outperform other models, especially regarding the trusted misclassification rate. This observation raises a central issue, i.e. it's difficult to establish a direct link between the accuracy or reliability of a model and the misclassification rate.

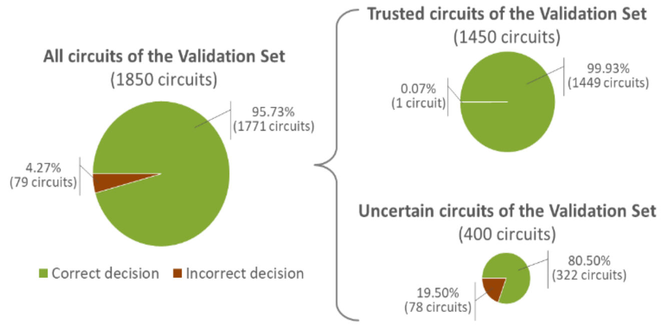

Fig.5. Repartition of misclassified instances using *MLR* model with 10 features

Furthermore, another important observation can be noticed in relation to the values achieved in terms of misclassification rate. Even if we have models with very good accuracy and reliability, more than 4% of the circuits suffer from an incorrect decision according to the classical *MR* definition, which can be considered as a poor performance. In contrast when looking only at circuits with a certain decision, a very good performance is achieved with a trusted misclassification rate that falls below 0.5%. There is a factor about 10 between the classical misclassification rate and the trusted one. This big difference indicates that most of misclassified circuits are located close to the test limit and within the uncertainty region; the classical misclassification rate fails to reveal this situation. To further illustrate this point, Figure 5 gives the repartition of misclassified instances using a *MLR* model built with at least 10 features. It clearly illustrates that, among the 79 circuits that have an incorrect decision for the classical computation of the misclassification rate, 78 of them are actually circuits located in the uncertainty region. Only 1 out of these circuits experiences a truly incorrect decision, which corresponds to a *T-MR* as low as 0.07%. More generally, this result illustrates the fact that, because it does not take into account the measurement uncertainty that affects the conventional specification test, the misclassification rate classically computed is a pessimistic metric and is should not be considered as truly representative of the indirect test efficiency.

## V. Conclusion

In this paper, we aimed at highlighting the issue of defining a pertinent metric able to assess the performance of various prediction models, which will lead us towards an automated choice of the best test strategy in a given indirect test context. Based on the case study results presented in the previous section, we can assert that there is an inconsistency on the model performance reflected by the various metrics. Therefore, we can observe the dire need of defining what is the most pertinent criterion to perform model selection. Solving this problem will allow us to have a more robust prediction model, in addition we may be able to use it to perform feature selection and even limit our feature space by applying a stopping criterion based on best model metric.

## Acknowledgment

This work has been carried out under the framework of PENTA-EUREKA project "HADES: Hierarchy-Aware and secure embedded test infrastructure for Dependability and performance Enhancement of integrated Systems".

## References

[1] P. N. Variyam et al., "Prediction of analog performance parameters using fast transient testing," IEEE Trans. On Computer-Aided Design of Integrated Circuits and Systems, Vol. 21, no. 3, pp. 349–361, 2002.

[2] S. Ellouz et al., "Combining internal probing with artficial neural networks for optimal RFIC testing", Proc. IEEE Int'l Test Conference (ITC), p.9, 2006.

[3] H. Ayari, et al., "Making predictive analog/RF alternate test strategy independent of training set size", Proc. IEEE Int'l Test Conference (ITC), p.9, 2012.

[4] L. Abdallah et al., "Sensors for built-in alternate RF test," Proc. IEEE European Test Symposium (ETS), pp. 49-54, 2010.

[5] M.J. Barragan, et al., "Improving the Accuracy of RF Alternate Test Using Multi-VDD Conditions: Application to Envelope-Based Test of LNAs", Proc. IEEE Asian Test Symposium (ATS), pp. 359-364, 2011.

[6] M.J. Barragan, G. Leger, "Efficient selection of signatures for analog/RF alternate test", Proc. European Test Symp. (ETS), p.6, 2013.

[7] A. Gómez-Pau, L. Balado, J. Figueras, "Quality metrics for mixed-signal indirect testing", Proc. IEEE Design of Circuits and Integrated Systems (DCIS), 2014.

[8] M.J. Barragan, G.Leger, "A Procedure for Alternate Test Feature Design and Selection", IEEE Design & Test, Vol. 32, pp. 18-25, 2015.

[9] S. Larguech, and al., "Efficiency evaluation of analog/RF alternate test: Comparative study of indirect measurement selection strategies", Microelectronics Journal,Vol. 46, n°. 11, pp. 1091-1102, 2015.

[10] H. Stratigopoulos, "Machine learning applications in IC testing", Proc. IEEE European Test Symposium (ETS), p.10, 2018.

[11] I. Guyon, A. Elisseeff, "An introduction to variable and feature selection", Journal of Machine Learning Research, Vol. 3, pp. 1157-1182, 2003.

[12] Z-H. Zhou, "Ensemble Methods: Foundations and Algorithms", Chapman and Hall/CRC, Machine Learning & Pattern Recognition Series, 1st Edition, 2012.

[13] H. El Badawi, "Use of ensemble methods for indirect test of RF circuits: can it bring benefits?", IEEE Latin Test Symp. (LATS), 2019.

[14] S. Larguech, and al., "A Framework for Efficient Implementation of Analog/RF Alternate Test with Model Redundancy", Proc. IEEE Computer Society Annual Symp. on VLSI, pp. 621-626, 2015.